

PATENT ABSTRACTS OF JAPAN

(11)Publication number :

2002-222110

(43)Date of publication of application : 09.08.2002

(51)Int.Cl.

G06F 12/00

G06F 15/177

(21)Application number : 2001-016503

(71)Applicant : HITACHI LTD

(22)Date of filing : 25.01.2001

(72)Inventor : TANAKA ATSUSHI

OBARA KIYOHIRO

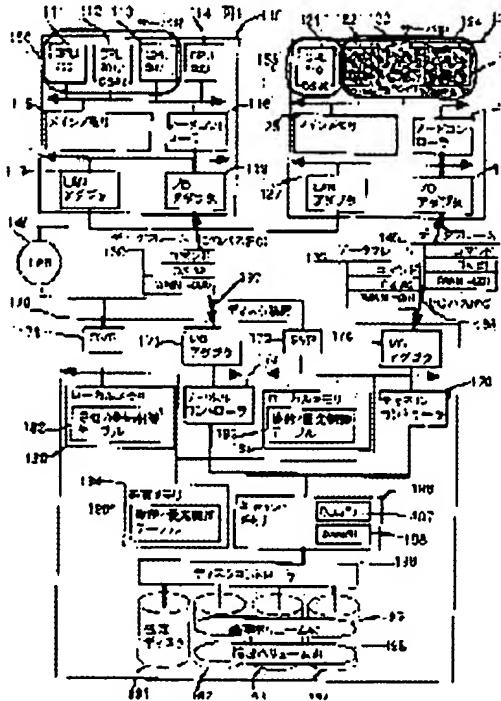
ODAWARA HIROAKI

(54) STORAGE SYSTEM AND VIRTUAL PRIVATE VOLUME CONTROLLING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To change an exclusive control method and a priority control method for improving security ability and optimizing performance of a system even when there are a plurality of operating systems in a server and a path to the disk device is shared between them by making discrimination between a plurality of operating systems and applications on the disc device side.

SOLUTION: In this virtual private volume controlling device, an identification number for identifying the operating system is given to the inside of a command issued by the operating system, and according to the identification number, exclusion and priority of access is controlled inside the disk device. In this way, transaction performance of the server and the disk device as a whole can be improved.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(51)Int.Cl. ⁷ G 0 6 F 12/00 15/177	識別記号 5 3 7 5 0 1 6 8 2	F I G 0 6 F 12/00 15/177	テマコード*(参考) 5 3 7 A 5 B 0 4 5 5 0 1 A 5 B 0 8 2 6 8 2 G
---	---------------------------------	--------------------------------	---

審査請求 未請求 請求項の数24 OL (全24頁)

(21)出願番号 特願2001-16503(P2001-16503)

(22)出願日 平成13年1月25日(2001.1.25)

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(72)発明者 田中 淳
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 小原 清弘
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(74)代理人 100075096
弁理士 作田 康夫

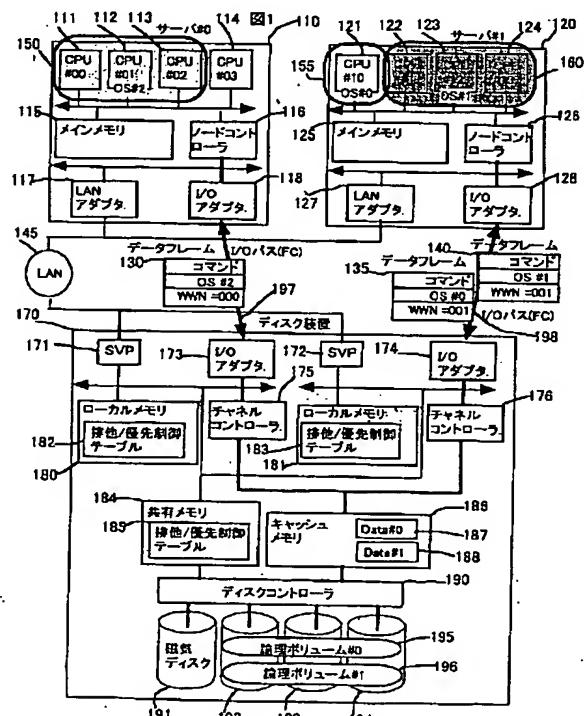
最終頁に続く

(54)【発明の名称】ストレージシステム及び仮想プライベートボリューム制御方法

(57)【要約】

【課題】 サーバ内に複数のOSが存在し、ディスク装置へのバスを共有していると、ディスク装置側ではこれら複数のOS、アプリケーションを区別することができない。そこで、複数のOS、アプリケーションを区別して、排他制御、優先制御方法を代えることでセキュリティ能力を向上させ、システムの性能最適化を図る。

【解決手段】 OSが発行するコマンドの内部にOSを識別する識別番号を付与し、ディスク装置内でその識別番号を元にアクセスの排他、優先処理を制御することで、サーバ及びディスク装置全体のトランザクション性能を高めることを特徴とする仮想プライベートボリューム制御装置とする。



【特許請求の範囲】

【請求項1】複数のOSを識別する識別番号を付与されたコマンドを受信し、前記識別番号を抽出し、前記コマンドに対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信することを特徴とするストレージシステム。

【請求項2】前記論理ボリュームは複数の磁気ディスク装置から構成されることを特徴とする請求項1記載のストレージシステム。

【請求項3】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項1または2記載のストレージシステム。

【請求項4】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項1乃至3記載のストレージシステム。

【請求項5】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項1乃至4記載のストレージシステム。

【請求項6】前記受信したコマンドの複数の識別番号の組み合わせに対応してレスポンスを返信することを特徴とする請求項1乃至5記載のストレージシステム。

【請求項7】複数のOSを有し、前記OSからのアクセス要求があると当該OSを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信し、前記サーバは前記レスポンスを受信することを特徴とする仮想プライベートボリューム制御方法。

【請求項8】前記サーバは前記OSを識別する識別番号をデータフレームに書き込み、当該データフレームをコマンドとして送信し、前記ディスク装置は当該データフレームを受信して前記識別番号を抽出することを特徴とする請求項7記載の仮想プライベートボリューム制御方法。

【請求項9】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項7または8記載の仮想プライベートボリューム制御方法。

【請求項10】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項7乃至8記載の仮想プライベートボリューム制御方法。

【請求項11】前記識別番号と前記コマンドのアクセス

の処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項7乃至10記載の仮想プライベートボリューム制御方法。

【請求項12】複数OSの少なくとも一つからアクセス要求があると当該OSを識別する識別番号を付与し、前記付与された識別番号をサーバ内に記憶し、前記識別番号に対応するレスポンスを受信し、当該レスポンスを前記OSへ返信することを特徴とするOS管理ソフトプログラム。

【請求項13】複数のOSおよびアプリケーションを識別する識別番号を付与されたコマンドを受信し、前記識別番号を抽出し、前記コマンドに対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信することを特徴とするストレージシステム。

【請求項14】前記論理ボリュームは複数の磁気ディスク装置から構成されることを特徴とする請求項13記載のストレージシステム。

【請求項15】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項13または14記載のストレージシステム。

【請求項16】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項13乃至15記載のストレージシステム。

【請求項17】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項13乃至16記載のストレージシステム。

【請求項18】前記受信したコマンドの複数の識別番号の組み合わせに対応してレスポンスを返信することを特徴とする請求項1乃至17記載のストレージシステム。

【請求項19】複数のアプリケーションおよびOSを有し、前記アプリケーションおよびOSからのアクセス要求があると当該アプリケーションおよびOSを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセスを処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号に付与して返信し、前記サーバは前記レスポンスを受信することを特徴とする仮想プライベートボリューム制御方法。

【請求項20】前記サーバは前記アプリケーションおよびOSを識別する識別番号をデータフレームに書き込み、当該データフレームをコマンドとして送信し、前記ディスク装置は当該データフレームを受信して前記識別番号を抽出することを特徴とする請求項19記載の仮想

プライベートボリューム制御方法。

【請求項21】前記受信したコマンドの識別番号に対応してアクセス処理の優先順位を変えることを特徴とする請求項19または20記載の仮想プライベートボリューム制御方法。

【請求項22】前記受信したコマンドのアクセスを処理するかまたはアクセスを拒否するかをあらかじめ設定した条件に基づき判断し、前記レスポンスを返信することを特徴とする請求項19乃至21記載の仮想プライベートボリューム制御方法。

【請求項23】前記識別番号と前記コマンドのアクセスの処理または拒否との関係を記憶したテーブルを有することを特徴とする請求項19乃至22記載の仮想プライベートボリューム制御方法。

【請求項24】複数アプリケーションの少なくとも一つからアクセス要求があると当該アプリケーションおよびOSを識別する識別番号を付与し、前記付与された識別番号をサーバ内に記憶し、前記識別番号に対応するレスポンスを受信し、当該レスポンスを前記アプリケーションおよびOSへ返信することを特徴とするアプリケーションおよびOS管理ソフトプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、サーバおよび記憶装置技術に関し、特に複数の記憶装置、CPUなどの構成要素を接続し、高いセキュリティ能力、高い応答性能、トランザクション数が要求される情報処理装置、記憶装置に有効な技術に関する。

【0002】

【従来の技術】複数のCPU、メモリとI/Oアダプタを持つサーバと大容量のディスク装置を組み合わせ、複数のユーザでそれら資源を共用する環境が現実になってくるに従って、各ユーザのセキュリティを守り、効率的に資源を使うことが要求されてきている。従来のサーバ及びディスク装置においては、論理ボリュームの排他制御を行うためにサーバに内蔵しているI/Oアダプタ毎に識別番号を持ちそれをストレージ側で区別する方式が採用されている。図2に従来のサーバ、ディスク装置構成例を示す。ここでサーバ#0(110)とサーバ#1(120)はそれぞれ複数のOS#0(155)～OS#2(120)を内蔵している。またI/Oアダプタ118、128を持ちストレージ170に接続している。ストレージ170にアクセスする場合データフレーム130、135、140をI/Oアダプタ118、128より発行する。データフレーム130、135、140の記述内容の中にはWorld Wide Name (WWN) (205、215、225)がありI/Oアダプタ118、128毎に固有の認識番号をそこに書きこむ。一方ディスク装置170ではI/Oアダプタ173、174を持ち上記データフレーム130、135、

140を受信する。ディスク装置170で内部制御を行うチャネルコントローラ175、176は受信したデータフレーム130、135、140から識別番号であるWWN205、215、225を抽出する。そしてチャネルコントローラ175、176はWWN205、215、225とローカルメモリ180、181にあらかじめ登録されている排他／優先制御テーブル182、183を比較することで論理ボリューム195、196に対するアクセスの排他／優先制御を行う。例えばサーバ#0(110)のOS#2(150)が論理ボリューム#0(195)を所有し、サーバ#1(120)のOS#1(160)が論理ボリューム#1(196)を所有していることを仮定する。ここで誤ってOS#1(160)が論理ボリューム#0(195)をアクセスした場合、このアクセスのデータフレーム140のWWN225には識別番号001があるので、チャネルコントローラ176は排他／優先制御テーブル183を参照し、識別番号001しかアクセスが許可されていないことがわかる。そこでチャネルコントローラ176はOS#1(160)に対してアクセスが拒否されたことを伝えれる。このような論理ボリュームの排他／優先制御を備えたディスク装置の技術については、特開2000-20447号に開示された「記憶サブシステム」等がある。【0003】一方サーバ内に複数の資源が存在する場合それらを複数のOSで分割して使う技術が存在する。この技術を用いることで、サーバのCPU、メモリ等を効率よく利用することが可能になってきている。これらの技術の一例としてLPAR(Logical Partitioning)がある。これを用いた場合、複数のOSを管理するソフトウェアプログラムとしてハイパバイザ430が存在することにより、動的にOSに、CPU、メモリ、I/Oデバイスを割り当てることが可能となる。例えば図2のサーバ#1(120)では、CPU#10～#13と4個あるので、LPARを用いることでOS#0(155)にはCPU#10を割り当て、OS#1(160)にはCPU#11、#12、#13(122、123、124)を割り当てることができる。メインメモリ125についても容量を適当な割合でOS#0(155)、OS#1(160)に割り当てることができる。I/Oアダプタが複数ある場合はOS#0(155)、OS#1(160)に割り当てることができるがこの例では1個しかないので、I/Oアダプタ128は両OSで共有することになる。このような技術については、特開平10-301795号に開示された「仮想計算機システム」等がある。また、特開2000-112804号には、複数のOSがファイルを共用して動作する仮想計算機システムにおいて、1つのOSが他のOSのデータファイルをアクセスすることを防止するため、OS番号が付されたI/O命令をディスク装置に対して発行することが記載されている。また、特開平4-

140844号には、OSが発行したI/O命令を解析するI/O解析手段を有し、I/O解析手段では引き取ったI/O命令にOS識別子を付加し、複数OSを同時動作させたときに排他制御することが記載されている。

【0004】

【発明が解決しようとする課題】以上に述べたように、WWN205、215、225をディスク装置170で識別することによって論理ボリューム195、196の排他／優先制御を行うことが可能となる。またLPARを用いることでサーバ内複数のOSに対して、CPU、メモリ、I/Oデバイスを効率よく割り当てることが可能になる。しかし一般にサーバ内CPU数の増加の割合に対して、I/Oデバイスの数はそれほど多く増加せず、CPU間で共有する場合が多い。これはコストを削減とI/Oデバイスが外部へのI/Fを持つことによる物理的な大きさのためである。このような状況で論理ボリュームの排他／優先制御を行う場合、以下のように排他／優先制御が出来なくなる問題がある。例えば、図2のサーバ#1(120)ではOS#0(155)とOS#1(160)が同じサーバ内に存在し、I/Oアダプタ128を共有して使うことになる。しかし両OSから発行するデータフレーム135、140のWWN215、225はI/Oアダプタ128を共有するため同じ識別番号“001”が付けられる。よってディスク装置170ではこの2個のデータフレームについて排他／優先制御は行うことができない。よってOS#0(155)が論理ボリューム#0(195)を所有、他のOSにアクセスすることを許可しない場合でも、上記理由によりOS#1(160)は論理ボリューム#0(195)にアクセスすることが可能となってしまう。本発明の主たる目的は、サーバ内の複数OSがI/Oデバイスとストレージシステムを共有する場合に、各OSに固有な識別情報をサーバ側で付加、ディスク装置側で抽出する手段をつけることにより各OS毎に所有する論理ボリュームの排他／優先制御を行う技術を改良することにある。本発明のさらに別の目的は、ディスク装置に排他／優先制御方法を記述したテーブルを設置し、複数の排他／優先制御方法から選択し実行する技術を提供することにある。本発明のさらに別の目的は、サーバからディスク装置に対して、各OS毎に複数の排他／優先制御方式をディスク装置に対して命令する技術を提供することにある。

【0005】

【課題を解決するための手段】上記問題を解決するために、複数のOSを有し、前記OSからのアクセス要求があると当該OSを識別する識別番号を付与し、前記付与された識別番号を有するコマンドを送信するサーバと、当該送信されたコマンドを受信するディスク装置とを備え、前記ディスク装置は前記識別番号を抽出し、前記抽出した識別番号に対応した論理ボリュームへのアクセス

を処理するかまたはアクセスを拒否するかに相当するレスポンスを前記識別番号を付与して返信し、前記サーバは前記レスポンスを受信する仮想プライベートボリューム制御方法とする。また、サーバにOS毎の識別番号をディスク装置アクセス時に発行するデータフレーム内に書き込む手段をおく。また、ディスク装置の中で上記OS毎の識別番号を抽出し、それを元にアクセスの排他／優先制御を行う手段をおく。さらにサーバからディスク装置にOS毎にアクセス可能な論理ボリュームまたは排他／優先制御情報を送る手段をおく。さらにディスク装置に上記OS毎にアクセス可能な論理ボリュームまたは排他／優先制御情報を格納するテーブルをおく。ここで、複数のOSとは、OSユーザーが異なる場合などの、仮想的に異なる複数のOSであればよく、同種のOS(例えばOS#0がWindowsNT[米国およびその他の国における米国Microsoft Corp.の登録商標；以下同じ]でOS#1がWindowsNTの場合、OS#1がUNIX[X/Open Company Limitedが独占的にライセンスしている米国ならびに他の国における登録商標；以下同じ]でOS#2がUNIXなど)であってもよい。

【0006】

【発明の実施の形態】(実施例1)以下、本発明に係る実施例1を図面に示しさらに詳細に説明する。図1は本発明の実施例1における仮想プライベートボリューム制御装置の全体構成図である。110、120はサーバであり、OS、アプリケーションを動かし、ディスク装置170に対してデータの読み出し、書き込み等の命令を発行する。111、112、113、114、121、122、123、124はCPUであり、サーバ#0、#1(110、120)に複数格納され、OS、アプリケーションの処理を並列に行う。115、125はメインメモリであり、CPUが直接使うデータを格納している。116、126はノードコントローラであり、CPUとI/OアダプタまたはLANアダプタ等の下位アダプタ間の転送を制御する。117、127はLANコントローラであり、CPU(111、112、113、114、121、122、123、124)とディスク装置170間の転送を制御する。図1の例では複数のCPUでメインメモリ、LANアダプタ、I/Oアダプタを共用することになる。145はLAN(Local Area Network)でありサーバとディスク装置間のデータを通す通信路である。このLAN145を使ってディスク装置170はサーバ110、120からの排他／優先制御に関する情報を入手する。170はディスク装置であり、サーバ#0、#1(110、120)の命令に従つてデータの読み出し、書き込みを行う。171、172はSVP(Service Processor)であり、LAN145を経由

してサーバ#0、#1（110、120）より送られるディスク装置170の制御情報を受信し、ディスク装置170の必要な制御情報を変更する。175、176はチャネルコントローラであり、サーバ#0、#1（110、120）からくるコマンドを解析し、必要なデータの検索、送受信を行う。180、181はローカルメモリであり、チャネルコントローラ175、176を制御する際に必要な制御情報を格納する。184は共有メモリであり、チャネルコントローラ175、176間の通信、制御を行う際に必要な制御情報を格納する。182、183、185は排他／優先制御テーブルであり、論理ボリューム等の排他／優先制御を行う際に必要な制御情報を格納する。詳しくは図6にて説明する。186はキャッシュメモリであり、サーバ#0、#1（110、120）が必要としているデータを格納するメモリである。ただしすべてのデータは格納されておらず、使用頻度の高いデータのみ格納されている。187、188はキャッシュメモリ186内に格納されたデータである。190はディスクコントローラであり、磁気ディスク191、192、193、194とキャッシュメモリ186間データの送受信制御を行う。191、192、193、194は磁気ディスクであり、データが最終的に格納されている記憶装置である。150、155、160はOSであり、アプリケーションからのアクセスの制御等を行う。195、196は論理ボリュームであり、OS#0（150）、OS#1（155）、OS#2（160）に割り当てられる記憶領域である。OS#0（150）、OS#1（155）、OS#2（160）から論理ボリューム#0（195）、#1（196）に対するアクセス命令はI/Oアダプタ118、128でコマンドおよび制御情報を変換され、データフレーム130、135、140内に書き込まれ、ディスク装置170へ転送される。データフレーム内の詳細については図3にて説明する。197、198はI/Oバスであり、サーバ#0、#1（110、120）とディスク装置170を接続しているデータの通信路である。この例ではファイバーチャネル（FC）を示しているが他の規格でも本発明の範囲内である。ここで、複数のOSとは、OSユーザーが異なる場合などの、仮想的に異なる複数のOSであればよく、同種のOS（例えばOS#0がNTでOS#1がNTの場合、OS#1がUNIXでOS#2がUNIXなど）であってもよい。

【0007】図3に本発明の実施例1におけるデータフレーム構造例を示す。この構造例はファイバーチャネルの標準規格のフォーマットを参考したものである。300はデータフレームの例であり、幾つかの部分から成り立つデータである。305はStart Of Frame識別子であり、データフレームの先頭をI/Oアダプタ118、128、173、174に伝える。310はフレームヘッダであり、データフレームのアドレス等を記述してい

る。詳細は後で述べる。315は、ペイロードであり上位プロトコルのデータ、コマンド等が格納されている。詳細は後で述べる。320はCRC（Cyclic Redundancy Check）であり、データフレーム300内のデータの誤りを発見し訂正を行う。325はEnd Of Frame識別子であり、データフレームの終わりをI/Oアダプタ118、128、173、174に伝える。次にフレームヘッダ310の構造例を示す。330は送信先ポートIDでありデータフレーム300が次に進みたい装置のポートIDを示す。335は送信元ポートIDでありデータフレームをディスク装置170に送信してきた装置のポートIDを示す。具体的にはWWNの情報がここに格納され、異なるI/Oアダプタからのデータフレームの排他／優先制御に用いられる。340は上位レイヤプロトコルであり、送信元と送信先がデータのやり取りを行う際に使うプロトコルの種類を示しており、たとえばSCSI等があげられる。345は上位レイヤシーケンスIDであり、上位プロトコルに従った通信におけるシーケンスの順番を示す。350はシーケンス内フレーム番号であり、シーケンス内にある複数のデータフレームの順番を示す。355は上位レイヤプロトコル送信先IDであり、最終的にデータフレーム300が到着する装置のポートIDを示す。次にペイロード内容としてSCSIを上位プロトコルに使った場合のリードコマンド構造例を示す。360はグループコードであり、コマンドの長さまたは標準のコマンドかベンダユニークなコマンドかの区別をする。365はコマンドコードでありこの例の場合はリードコマンドに対応するコードが入る。370は論理ボリューム番号（LUN）であり例えば磁気ディスク191～194に設定された論理ボリューム#0（195）、#1（196）を示している。375はデータ等が存在する場所のアドレスであり、例えば論理ボリューム#0（195）、#1（196）内のアドレスを示している。380はLPAR-IDを示しており、具体的にはハイババイザ430の指定した各OSの識別番号が入る。385は転送データ長であり、リードする範囲を示す。390はコントロールバイトであり、データフレーム300またはペイロード315が扱える容量に合わせるために付け加える空白なデータである。なお上記例以外のフォーマットを使ってWWN、LPAR-IDをフレームに内蔵させた場合でも、本発明の範囲であることは自明である。図4に本発明の実施例1におけるソフトウェア構造例とデータ転送成功例を示す。ここでサーバ#0（110）のみ説明する。410、411、412、413はミドル／アプリであり、ユーザの処理を直接受けるプログラムである。420、421、422、423はOSであり、ミドル／アプリ410、411、412、413からの要求を調整、処理を行う。430はハイババイザであり、サーバ#0（110）内のCPU、メインメモリ、I/Oデバイスを複数のOS420、421、4

22、423に分割して利用させる。また各OS420、421、422、423の識別番号となるLPAR-ID450、451、452を管理、設定する。440、441、442はデバイスドライバであり、ディスク装置170へのアクセスが必要な場合、コマンド、データ、WWN、LPAR-ID等をデータフレーム130に設定してディスク装置170に送信する。455は制御フレームであり、サーバ#0(110)からディスク装置170の制御方式等を変更したい場合にこの中に設定してディスク装置170にあるテーブルセットプログラム470に送信する。制御フレームの詳細については図7にて説明する。460、465はディスク装置170のμプログラム(μP)であり、ディスク装置170のI/Oアダプタ173、174、チャネルコントローラ175、176、キャッシュメモリ186を制御する。ディスク装置170のμプログラム460は、I/Oアダプタでフレームの受信、解析を制御するI/OアダプタμPと、チャネルコントローラでコマンドの解析、排他／優先制御、コマンド実行を制御するチャネルコントローラμP(462)に分かれている。さらにチャネルコントローラμP(462)ではデータフレーム130内に書きこまれている、WWN、LPAR-IDを抽出して、排他／優先制御テーブル182、185の内容と比較することにより、I/Oアクセスの排他／優先制御を実行する。排他／優先制御の詳細は図12、12、13、14で説明する。470はテーブルセットプログラムであり、サーバ#0(110)から来た制御フレーム455を受信し、内容をローカルメモリ180または共有メモリ184内の排他／優先制御テーブル182、185に格納する。図4のようにデータフレーム130のLPAR-IDがローカルメモリ180、または共有メモリ184にある排他／制御テーブル182、185内で論理ボリューム#0(195)へのアクセスを許可されている場合、アクセスの結果は以下のようになる。アクセスが許可されているので、論理ボリューム#0(195)内の必要なデータをリードし、フレーム136にLPAR-IDといっしょに載せてサーバ#0(110)に送る。さらにデータ転送が正しく処理したことステータスとして、フレーム137にLPAR-ID等といっしょに載せてサーバ#0(110)に送る。これらのフレーム136、137を受け取ることによりサーバはデータアクセスを完了する。図4の様なソフトウェア構造においてハイパバイザ430がデバイスドライバ440、441、442にLPAR-IDを設定する方式は2通りある。一つは初期設定時にデバイスドライバ440、441に設定する方式である。OS420、421とデバイスドライバ#0(440)、#1(441)が一対一に対応している場合はOSにLPAR-IDが割り当てられた時に各デバイスドライバに来るI/OのLPAR-IDも決まっているので、初期設

定時にハイパバイザ430よりOS420、421、デバイスドライバ#0(440)、#1(441)がそれぞれLPAR-ID#0、#1(450、451)を受け取り設定する。I/Oアクセス発生時にはデバイスドライバ#0、#1(440、441)は設定されたLPAR-ID#0、#1(450、451)をデータフレーム130に設定する。2種類目はI/Oアクセス毎にデバイスドライバ#2(442)にLPAR-ID#2/3(452)を連絡する方式である。OS422、423はデバイスドライバ#2(442)を共有しているので実際にI/Oアクセスが発生するまでどちらのOSのI/Oアクセスなのか不明である。そのためOSがI/Oアクセスをデバイスドライバ#2(442)に発行する際に、前もってハイパバイザより設定されたLPAR-ID#2/3(452)を同時にデバイスドライバ442に送信する。受信したデバイスドライバ#2(442)はそのLPAR-ID#2/3(452)をデータフレーム130に設定する。以上のようにしてデータフレーム130にLPAR-ID#0、#1、#2、#3(450、451、452)が設定されると、同じI/Oアダプタ118を共有している複数のOS(420、421、422、423)からI/Oアクセスが発生した場合でも、LPAR-ID#0、#1、#2、#3(450、451、452)が異なるため、上で述べた様にストレージ側でこれを識別することで排他／優先制御を行うことが可能になる。図5に本発明の実施例1におけるソフトウェア構造例とデータ転送失敗例を示す。図5のようにデータフレーム130のLPAR-IDがローカルメモリ180、または共有メモリ184にある排他／制御テーブル182、185内で論理ボリューム#0(195)へのアクセスを許可されていない場合、アクセスの結果は以下のようになる。アクセスが許可されていないので、データ転送が正しく処理できずエラーとなる。エラーが発生したことをステータスとして、フレーム138にLPAR-ID等といっしょに載せてサーバ#0(110)に送る。このフレーム138を受け取ることによりサーバは次に行うべきエラー処理を判断することが可能になる。一般的にはこの後サーバ側がディスク装置のエラー情報をアクセスして次の処理を決定する。図4、5で示したサーバへのフレーム137、138の返信は本発明のひとつ目の例であり、排他／優先制御に関する他の情報を返信することが可能なことは明白である。図6に本発明の実施例1における排他／優先制御テーブル例を示す。500は例として論理ボリューム#0(195)の排他／優先制御テーブルを示している。505、510、515、520は論理ボリューム#0(195)アクセスする可能性があるWWNを示す。525、530、535、540は各WWNを持つI/Oアダプタからアクセスする可能性のあるOSのLPAR-IDを示す。この表で“○”で示されているWW

N、L P A R - I Dは論理ボリューム# 0 (195)にアクセスが可能である。逆に“×”の場合論理ボリューム# 0 (195)にアクセスは排他される。この制御方式は図13に詳細を説明する。また排他制御# 2と書いてあった場合は図14で詳細を説明する排他制御方式に従う。また優先制御と書いてあった場合は図15で示す優先制御方式に従う。ここで他の排他／優先制御を用いる場合でもこの排他／優先制御テーブル500に登録しておけば実現可能であることは自明である。排他／優先制御テーブルは論理ボリューム毎に存在し、それらの集合はローカルメモリ180、181、共有メモリ184内に存在する(排他／優先制御テーブル182、183、185)。そしてチャネルコントローラ175、176はこれらの表を参照して排他／優先制御を行う。これらの排他／優先制御テーブルはL A N 1 4 5を通して、サーバ# 0、# 1 (110、120)から送られる制御フレーム455をテーブルセットプログラム470が受信し、生成、変更することによってできる。制御フレーム455の詳細については図7で説明する。図7に本発明の実施例1における制御フレーム構造例を示す。制御フレーム構造例についてはI P、U D Pのプロトコルを参考にしたものである。またメッセージ構造例についてはS N M P (S i m p l e N e t w o r k M a n a g e m e n t P r o t o c o l)を参考にしたものである。ローカルネットワークヘッダ605には、L A N通信に必要なアドレス、データ長等の制御情報が格納されている。I Pヘッダ610にはI Pを使った通信に必要なアドレス、データ長等の制御情報が格納されている。U D Pヘッダ615にはU D Pを使った通信に必要なアドレス、データ長等の制御情報が格納されている。ここで取り上げた以外のプロトコルを使っても本発明の範囲であることは自明である。620はメッセージであり詳細については後で説明する。625はローカルネットワークトレーラであり、データのエラーチェックのコード、データの終端のコード等が記述されている。次にメッセージ620の構造例を示す。630はコマンドフィールドでありこのメッセージを使ってディスク装置が行うべきことを示している。この例ではパラメータを排他／優先制御テーブルにセットする命令が入っている。635はデータ長フィールドであり、以下のデータフィールドが何個続くかを示している。この例では6個続くことを示している。640～665まではデータフィールドであり、排他／優先制御テーブルに設定すべきパラメータが記述されている。この例では、論理ボリューム# 2 (640)のL A P R - I D # 2 (645)かつWWN# 1 (650)のアクセスに関しては排他制御# 2 (655)を行い、その制約条件は最大排他処理時間(660)と最大排他処理I／O数(665)であることを示している。データフィールドにはこのほかに優先制御を行う際の優先順位等が入ることも考えられる。

図8に本発明の実施例1におけるサーバV P V制御初期化のフロー示す。ここではサーバ# 0 (110)側仮想プライベートボリューム(V P V)制御の初期化を行う。ステップ700からサーバV P V制御初期化が始まる。ステップ705ではハイパバイザ430の初期化をまず行う。ステップ710ではL P A R - I D # 0 (450)、# 1 (451)、# 2 / 3 (452)を各L P A R (O S)に割り当てる。ステップ715では各L P A R内の特定エリアにI Dを格納する。ステップ720ではO S 4 2 0、4 2 1、4 2 2、4 2 3とデバイスドライバ440、4 4 1、4 4 2を初期化して立ち上げる。ステップ725ではO S 4 2 0、4 2 1、4 2 2、4 2 3間でデバイスドライバ440、4 4 1、4 4 2を共用しているかを調べる。もし共有していない(O S 4 2 0、4 2 1の場合)デバイスドライバ440、4 4 1が各L P A Rが格納しているL P A R - I D # 0 (450)、# 1 (451)を取得する。デバイス440、4 4 1、4 4 2を共有していれば(O S 4 2 2、4 2 3の場合)、O S 4 2 2、4 2 3がL P A R - I D # 2 / 3 (452)を取得する。ステップ740ではL P A R - I D # 0 (450)、# 1 (451)、# 2 / 3 (452)、論理ボリューム# 0、# 1 (195、196)排他／優先制御情報をディスク装置170に対して制御フレーム455を使って送信する。そしてステップ745でサーバV P V制御初期化が終了する。図9に本発明の実施例1におけるサーバ側I／O処理(デバイスドライバ非共有)のフローチャート示す。ここでは初期化終了後通常のI／O処理のフローを示す。またデバイスドライバはO S間で共有しない。ステップ800からサーバ側I／O処理が開始される。ステップ805ではO S 4 2 0、4 2 1にI／O処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ805に戻る。I／O処理が来ていればステップ810に進み、O S 4 2 0、4 2 1がデバイスドライバ440、4 4 1用のシステムコールを起動する。ステップ815ではデバイスドライバ440、4 4 1が起動する。ステップ820ではデバイスドライバ440、4 4 1がデータフレーム130に自分のL P A R - I D # 0 (450)、# 1 (451)を埋め込む。ステップ830ではディスク装置170からデータフレームに対するステータスが返送されたかどうかを調査する。返送していなければ、そのままステップ830に戻る。ステータスが返送された場合、ステップ835に進み、I／Oステータスが正常終了したかどうかを調べる。もし正常終了した場合はステップ805に戻る。異常終了した場合、ステップ840に進みエラー原因をディスク装置170に問い合わせる。ステップ845では問い合わせた結果を見て、サーバ# 0 (110)のエラー処理を行う。ステップ850ではI／O処理の回復は不可能かどうか調べる。もし回復が可能ならばステップ805に進む。不可能ならばステップ

855に進みサーバ側I/O処理を終了する。図10に本発明の実施例1におけるサーバ側I/O処理(デバイスドライバ共有)のフローチャートを示す。ここでは初期化終了後通常のI/O処理のフローを示す。またデバイスドライバはOS間で共有する。ステップ900からサーバ側I/O処理が開始される。ステップ905ではOS422、423にI/O処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ905に戻る。I/O処理が来ていればステップ910に進み、OS422、423がデバイスドライバ442用のシステムコールを起動する。ステップ915ではOS422、423がデバイスドライバ442に対して自分のLPAR-ID#2/3(452)を連絡する。ステップ920ではデバイスドライバ442がデータフレーム130に連絡されたのLPAR-ID#2/3(452)を埋め込む。ステップ930以降は図9のステップ830以降と同じなので以下説明を省略する。図11に本発明の実施例1におけるディスク装置VPV制御初期化のフローチャートを示す。ここではディスク装置170側仮想プライベートボリューム(VPV)制御の初期化を行う。ステップ1000からディスク装置VPV制御初期化を開始する。ステップ1005ではディスク装置170の初期化を行う。ステップ1010ではVPV制御μプログラム立ち上げ、排他/優先制御テーブル182、183、185の初期化を行う。ステップ1015ではサーバ#0(110)より制御フレーム455を通して、LPAR-ID、論理ボリューム、排他/優先制御に関する情報を受信する。受信なければ、ステップ1015に戻る。受信した後、ステップ1020では排他/優先制御テーブル182、183、185の設定を行う。そしてステップ1025でディスク装置VPV制御初期化が終了する。図12に本発明の実施例1におけるディスク装置側I/O処理フローチャートを示す。ここでは初期化終了後ディスク装置170における通常のI/O処理のフローを示す。ステップ1100よりディスク装置側I/O処理が開始される。ステップ1105ではI/Oアダプタ173、174にI/O処理が来ているかどうかを調べる。もし来ていなければ、そのままステップ1105に戻る。I/O処理が来ていればステップ1110に進み、I/Oアダプタμプログラム461を起動する。ステップ1115ではチャネルコントローラμプログラム462を起動する。ステップ1120ではコマンドに埋め込まれたLPAR-ID380を抽出する。ステップ1125ではI/O処理対象の論理ボリューム370の排他/優先制御テーブル500を参照する。ステップ1130ではI/OのLPAR-ID380、WWN335を参照してそれが排他制御対象かどうか調べる。排他制御対象でなければ、ステップ1150の優先制御処理に進む。ステップ1150については図15で詳細を説明する。排他制御対象ならばステップ1

135に進む。ステップ1135については図13、13で詳細を説明する。ステップ1150の優先制御処理の後、ステップ1135で通常処理と判定された場合はステップ1155にてI/Oアクセス処理のキューに入つてキャッシュメモリ186または磁気ディスク191、192、193、194のI/O処理を待つ。ステップ1135で排他処理と判定された場合はステップ1140に進む。ステップ1160ではキューに入れたI/Oの処理が正常に終わったかどうかを調べる。正常終了した場合はステップ1145へ進む。異常終了した場合はステップ1140へ進む。ステップ1140ではエラーメッセージを作成してディスク装置170内に保存する。ステップ1145はデータまたはステータスマップを作成して、サーバに返答する。その後ステップ1105に戻る図13に本発明の実施例1における第一の排他制御処理例のフローチャートを示す。ここでは、図12のステップ1135の具体的な制御方式として、一番単純な排他制御のフローを示す。ステップ1200より排他制御処理#1が始まる。ステップ1205ではアクセスが排他されるべきかどうかを調べる。排他ならばステップ1215の排他処理へ進む。排他でなければステップ1205の通常処理へ進む。図14に本発明の実施例1における第二の排他制御処理例のフローチャートを示す。ここでは、図12のステップ1135の具体的な制御方式として、処理時間、処理I/O数を条件とした、制限付き排他制御のフローを示す。ステップ1300より排他制御処理#2が始まる。ステップ1305ではそれまでに排他処理を行った時間が、あらかじめ設定してある最大排他処理時間(660)より少ないか調べる。少なければステップ1310へ進む。同じく、大きければステップ1335の通常処理へ進む。ステップ1310では排他処理時間を加算する。ステップ1315ではそれまでに排他処理を行ったI/O数が、あらかじめ設定してある最大排他処理I/O数(665)より少ないか調べる。少なければステップ1320に進む。同じく、大きければステップ1335の通常処理へ進む。ステップ1320では排他処理I/O数の加算を行う。ステップ1325ではアクセスが排他されるべきかどうかを調べる。排他ならばステップ1330の排他処理へ進む。排他でなければステップ1335の通常処理へ進む。図15に本発明の実施例1における優先制御処理例のフローチャートを示す。ここでは排他制御を行わず、処理の順番を変更するフローを示す。この例はSCSIの標準規格のタグキューイングを参考にしたものである。ステップ1400から優先制御処理が行われる。ステップ1405では処理するI/Oの処理順番を決めるためのポインタをキューの先頭にセットする。ステップ1410では処理するI/Oの優先順位を読み出す。ステップ1415ではキュー内のポインタの位置にI/Oがあるかどうかを調べる。なければ

ステップ1430へ進む。I/Oがあればステップ1420に進みポインタ位置にあるキューネI/Oの優先順位を読み出す。ステップ1425ではキューネI/Oの優先順位が処理するI/Oの優先順位より高いかどうかを調べる。優先順位高ければステップ1435に進み、ポインタを1加算して、ステップ1415に戻る。優先順位が低ければステップ1430へ進む。ステップ1430では処理するI/Oの順番を現時点でセットされたポインタの位置に決定する。ステップ1440では通常処理へ進む。なお以上のフロー例以外の方式を使った排他/優先制御処理も本発明の範囲であることは自明である。以上図1から図5までに示した実施例1によれば、LPAR-IDをデータフレームに入れてディスク装置に送信し、受信したディスク装置側で排他制御を行うことにより、I/Oアダプタを共有する複数のOSに関しても排他制御を行うことが可能となる。またディスク装置側でLPAR-ID、WWNを考慮した排他/優先制御テーブルを持つことにより、サーバ間をまたがったOS等に関してもディスク装置側で統一的に排他/優先制御を行うことが可能となる。また排他/優先制御方式も複数の方式を選択できるようにしているので、柔軟な排他制御、性能の最適化が可能となる。

(実施例2)以下、本発明に係るの実施例2を図面に示しさらに詳細に説明する。

【0008】図16に本発明の実施例2におけるソフトウェア構造例を示す。実施例1では排他/優先制御はディスク装置170で行ったが、単体サーバ#0(110)の内部だけに制御を限定し、サーバ#0(110)、ハイパバイザ1500の処理能力が高い場合は、その他にハイパバイザ1500行うことも可能である。この場合、すべてのI/Oアクセスはハイパバイザ1500で受け取り、ハイパバイザ内に格納されているLPAR-ID1510とその優先/排他制御テーブル1550を用いることで、排他/優先制御を行う。アクセスが許可された場合I/Oアクセスはそのままディスク装置170に送信され、アクセスが許可されない場合はそのままエラーでOS(420)に返される。以上図16に示した実施例によればディスク装置170、データフレーム140に変更を加えることなく、OS間の排他/優先制御を行うことが可能になる。

(実施例3)以下、本発明に係るの実施例3を図面に示しさらに詳細に説明する。

【0009】図17に本発明の実施例3におけるソフトウェア構造例を示す。実施例1、2では排他/優先制御を行う単位は、LPAR-IDを持つOS単位で行っていた。しかしユーザによっては、アプリケーション単位で排他を行う必要がでてくる。同一OS内のアプリケーションならば、OSが排他/優先制御を行うことになるが、複数OSにあるアプリケーション間で行うためには実施例1、2と同様にディスク装置460またはハイパバイザ

430で排他/優先制御を行うことになる。たとえばアプリケーション#1(1605、1615以下アプリ#1と呼ぶ)のデータアクセスは優先したい場合、アプリ#1(1605、1615)のアクセスが発生時にアプリケーション毎にあるID(APL-ID)をLPAR-IDと同様にデータフレーム1635につけ、ディスク装置460に送信する。ディスク装置460側では、前もって受信した制御フレーム1620で伝えられ排他/優先制御テーブル1625、1630に登録された、アプリケーション毎の排他/制御情報を元に排他/優先制御を行う。アクセス結果はデータ1640、ステータス1645によってサーバ#0(110)に報告される。この結果として、LPAR-ID#0(450)、LPAR-ID#2(452)によってアクセスが排他されていない限り、アプリ#1のデータアクセスはOSにかかわらず優先される。

【0010】図18に本発明の実施例3における排他/優先制御テーブル例を示す。実施例3ではアプリ毎に排他/優先制御を判定するために、図6に比較してテーブルが一種類増えることになる。1625-aは排他/優先制御テーブル例(WWN-LPAR関係)で図6の排他/優先制御テーブル500と同様な形式である。1625-bは排他/優先制御テーブル例(WWN-APL関係)であり、各WWN(1700~1715)に対するAPL(1720~1735)の排他/優先制御の有無を示してある。

【0011】図19に本発明の実施例3におけるデータフレーム構造例を示す。基本的には図3のデータフレーム構造と同じであるが、ペイロード315内にアプリケーションに関するIDとして、APL-ID(395)が付け加えられている。

【0012】図20に本発明の実施例3における制御フレーム例を示す。基本的には図7の制御フレーム構造と同じであるが、メッセージ620内のデータフィールドにアプリケーションに関するIDとして、APL-ID(670)が付け加えられている。以上図17に示した実施例によれば、図13~14で示した排他/優先制御処理のフローチャートと同様な方式で、異なるOSに存在するアプリケーション間の排他/優先制御を行うことが可能になる。

【0013】

【発明の効果】本発明によれば、複数OSがI/Oデバイスを共有してディスク装置をアクセスするとき、OS、アプリケーション毎に付与された認識番号をディスク装置が認識できるようになる。そのため、ディスク装置側でのOS、アプリケーション間のデータの排他制御が可能となりセキュリティを高めることができる。さらに、アクセスの優先制御も行うことが可能になり仮想プライベートボリュームシステム性能の最適化が図れる。

【図面の簡単な説明】

【図1】本発明の実施例1における全体構成図を示す。

【図2】従来の排他制御方式内蔵記憶装置の構成図を示す。

【図3】本発明の実施例1におけるデータフレーム構造例を示す。

【図4】本発明の実施例1におけるソフトウェア構造例とデータ転送成功例を示す。

【図5】本発明の実施例1におけるソフトウェア構造例とデータ転送失敗例を示す。

【図6】本発明の実施例1における排他/優先制御テーブル例を示す。

【図7】本発明の実施例1における制御フレーム例を示す。

【図8】本発明の実施例1におけるサーバVPI制御初期化のフローを示す。

【図9】本発明の実施例1におけるサーバ側I/O処理(デバイスドライバ非共有)のフローチャートを示す。

【図10】本発明の実施例1におけるサーバ側I/O処理(デバイスドライバ共有)のフローチャートを示す。

【図11】本発明の実施例1におけるディスク装置VPI制御初期化のフローチャートを示す。

【図12】本発明の実施例1におけるディスク装置側I/O処理フローチャートを示す。

【図13】本発明の実施例1における第一の排他制御処理のフローチャートを示す。

【図14】本発明の実施例1における第二の排他制御処理のフローチャートを示す。

【図15】本発明の実施例1における優先制御処理のフ

ローチャートを示す。

【図16】本発明の実施例2におけるソフトウェア構造例を示す。

【図17】本発明の実施例3におけるソフトウェア構造例を示す。

【図18】本発明の実施例3における排他/優先制御テーブル例を示す。

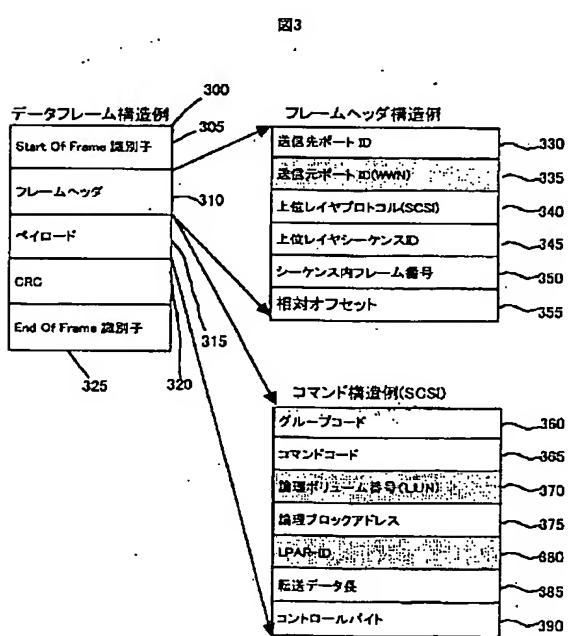
【図19】本発明の実施例3におけるデータフレーム構造例を示す。

【図20】本発明の実施例3における制御フレーム例を示す。

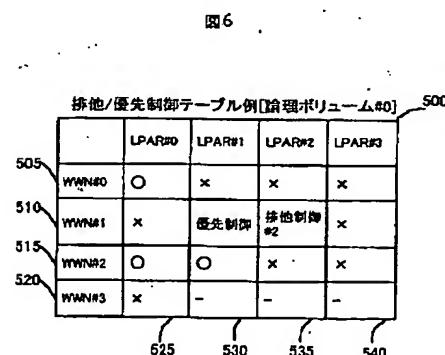
【符号の説明】

110、120…サーバ、111、112、113、
114、121、122、123、124…CPU、
115、125…メインメモリ、116、126…
ノードコントローラ、117、127…LANアダプタ、
118、128、173、174…I/Oアダプタ、
130、135、140…フレーム、145…
LAN、150、155、160…OS、170…
ディスク装置、171、172…SVP、175、1
76…チャネルコントローラ、180、181…ロ
ーカルメモリ、182、183、185…排他/優先
制御テーブル、184…共有メモリ、186…キ
ッシュメモリ、187、188…Data、190…
ディスクコントローラ、191、192、193、1
94…磁気ディスク、195、196…論理ボリュ
ーム、197、198…I/Oバス。

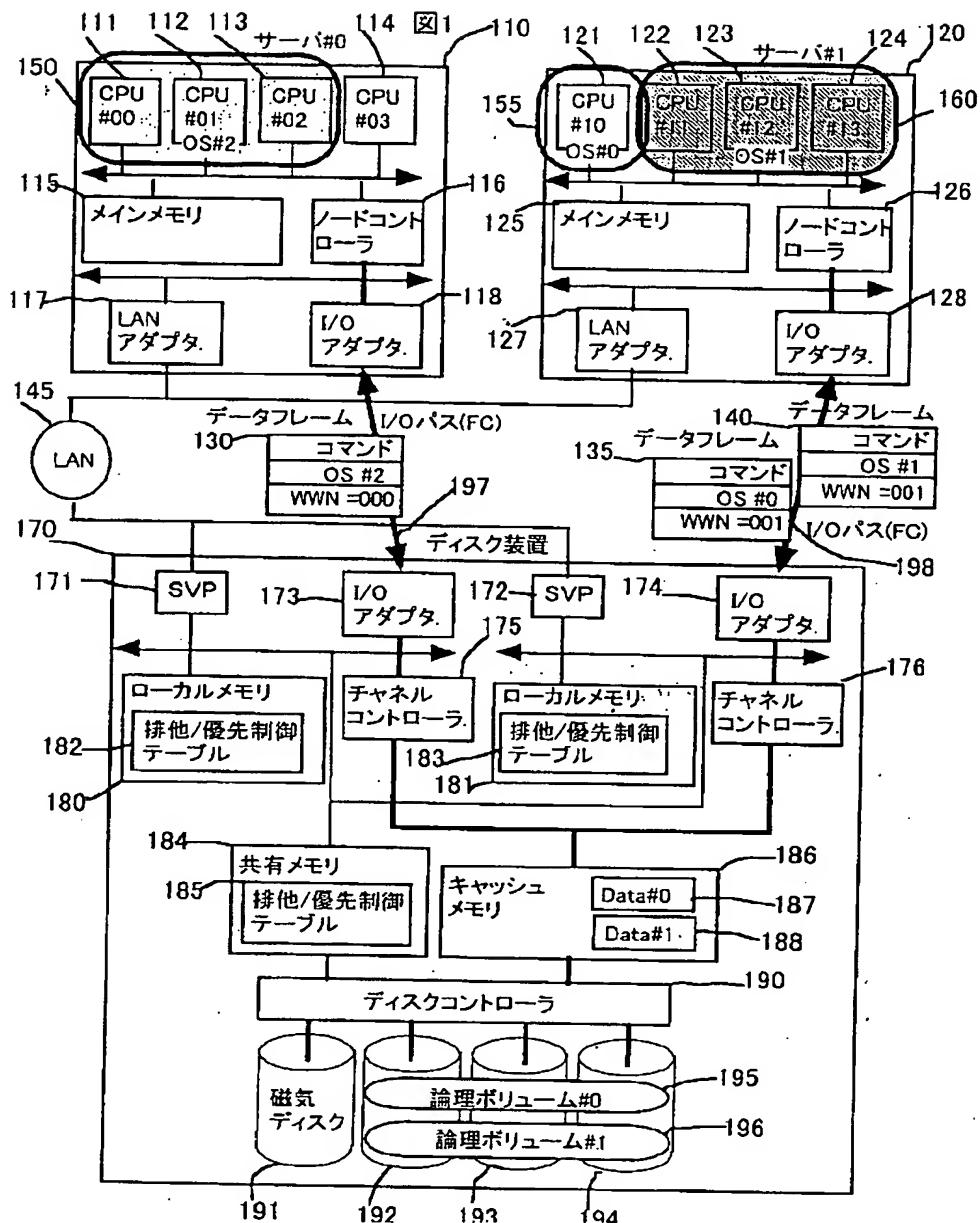
【図3】



【図6】

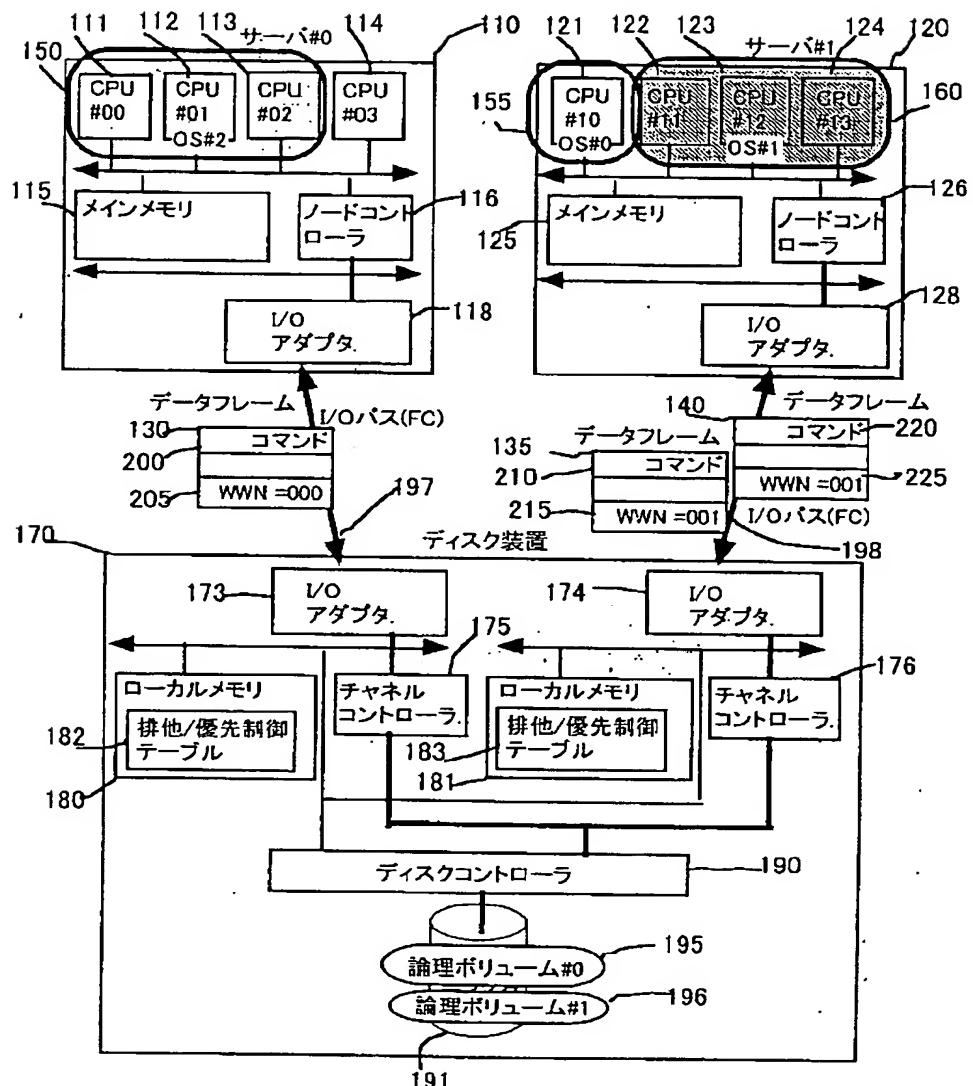


【図1】



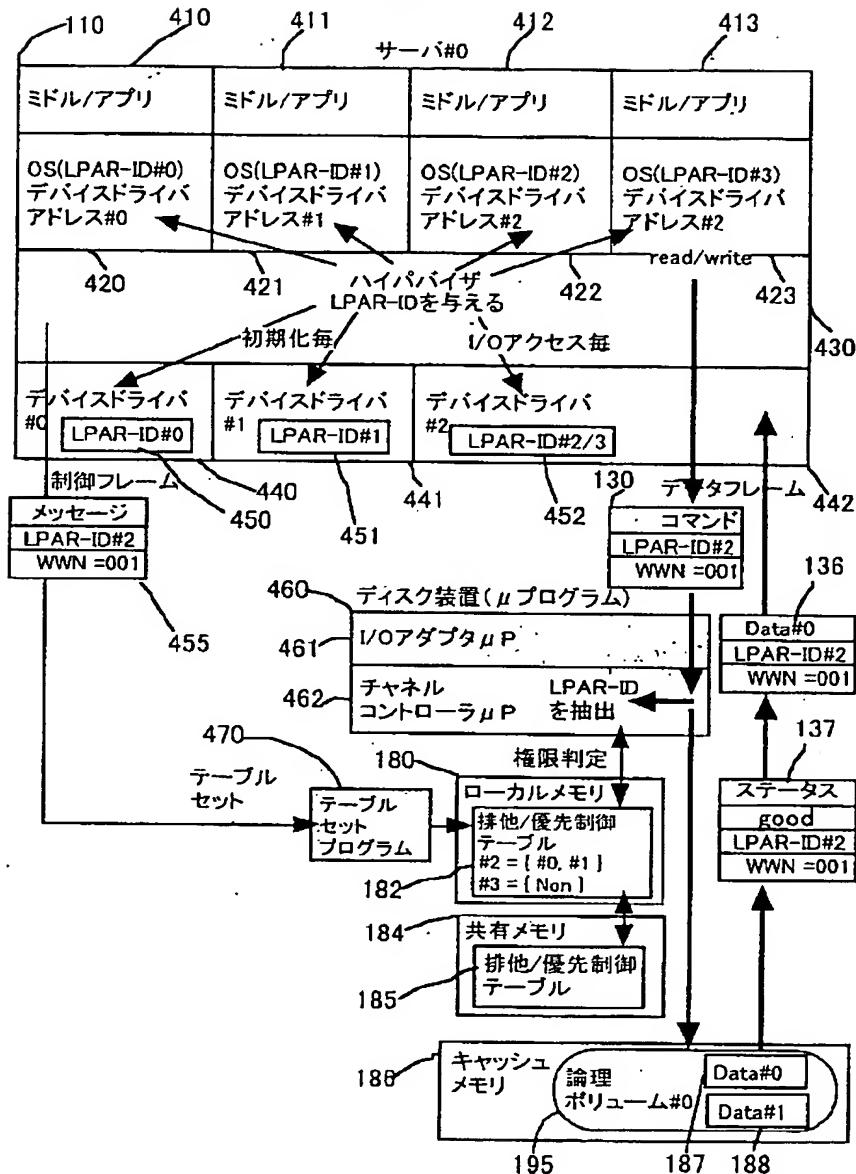
【図2】

図2



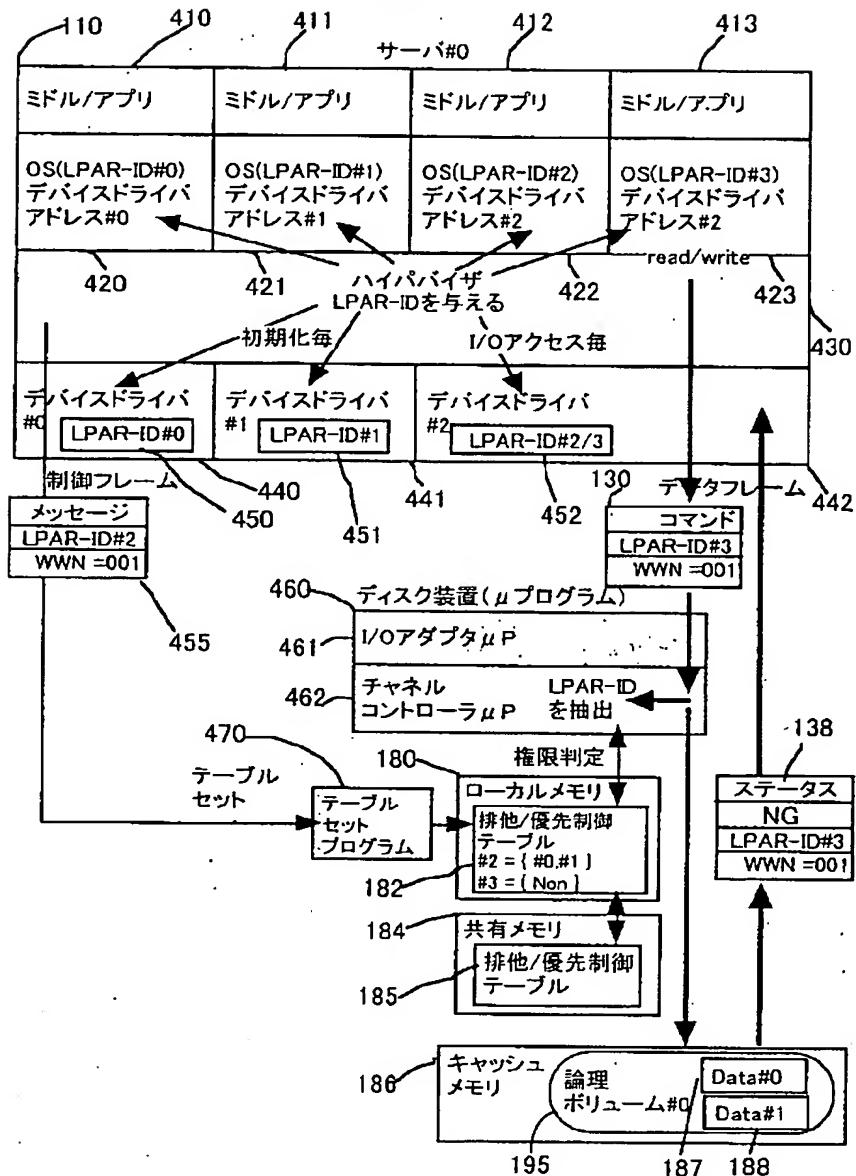
【图 4】

4



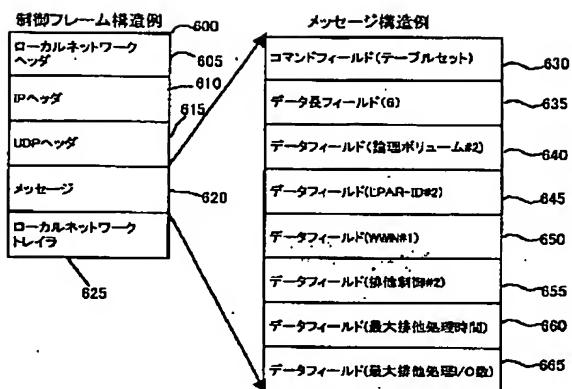
【図5】

図5



【図7】

図7



【図18】

図18

図18: 排他/優先制御テーブル例 [論理ボリューム#0] (WWN-LPAR関係)

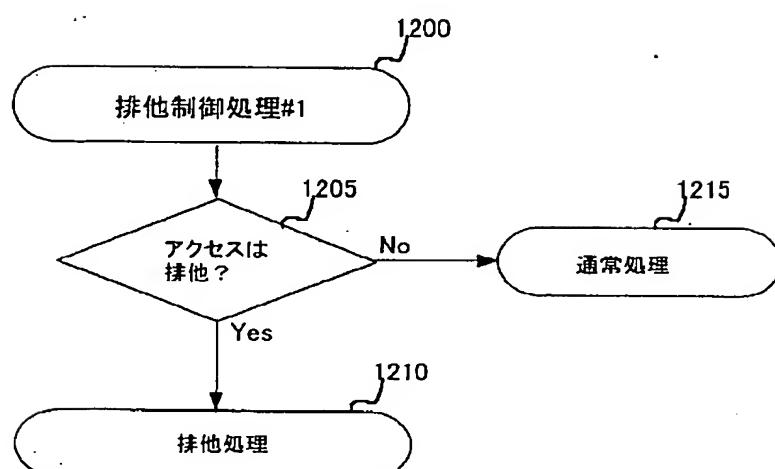
	LPAR#0	LPAR#1	LPAR#2	LPAR#3
WWN#0	○	×	○	×
WWN#1	×	優先制御#2	排他制御#2	×
WWN#2	○	○	×	×
WWN#3	×	-	-	-

図18: 排他/優先制御テーブル例 [論理ボリューム#0] (WWN-APL関係)

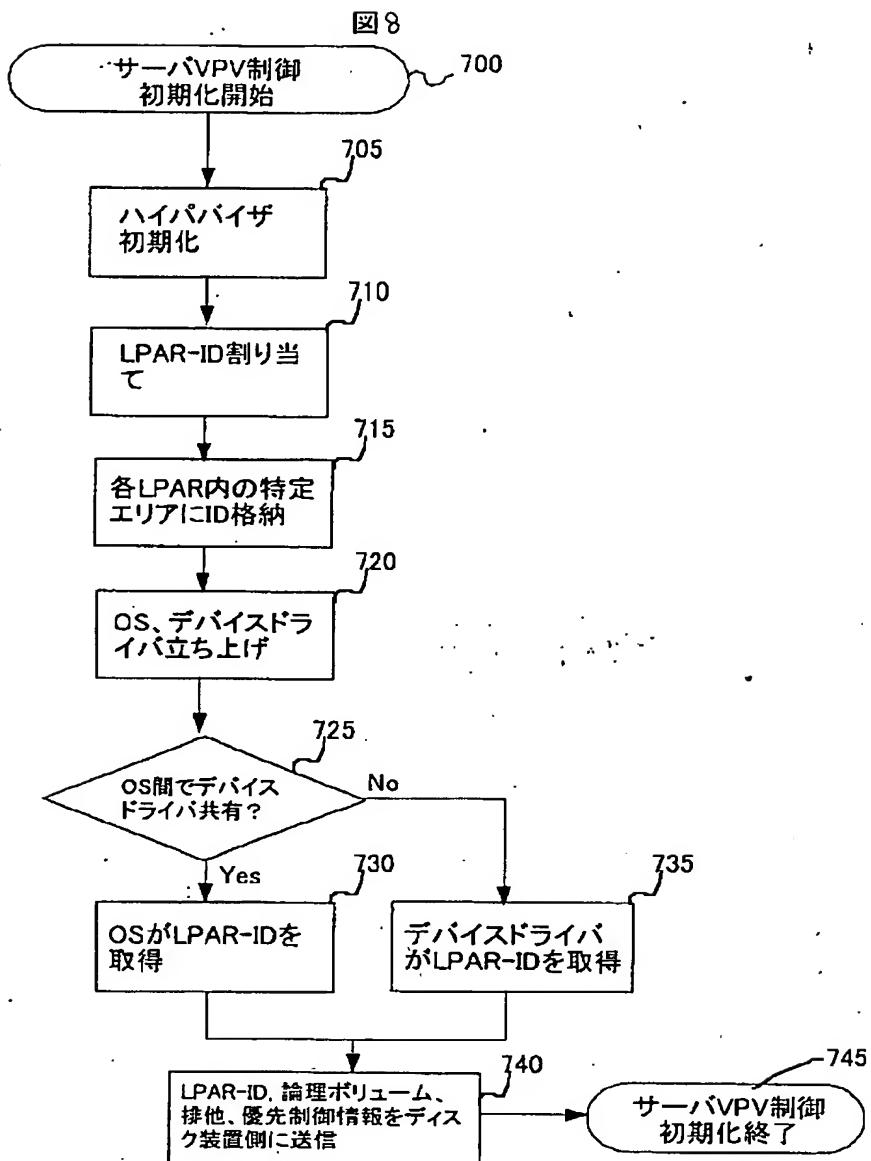
	APL#0	APL#1	APL#2	APL#3
WWN#0	×	○	○	×
WWN#1	×	優先制御#2	排他制御#2	×
WWN#2	×	○	×	×
WWN#3	×	-	-	-

【図13】

図13

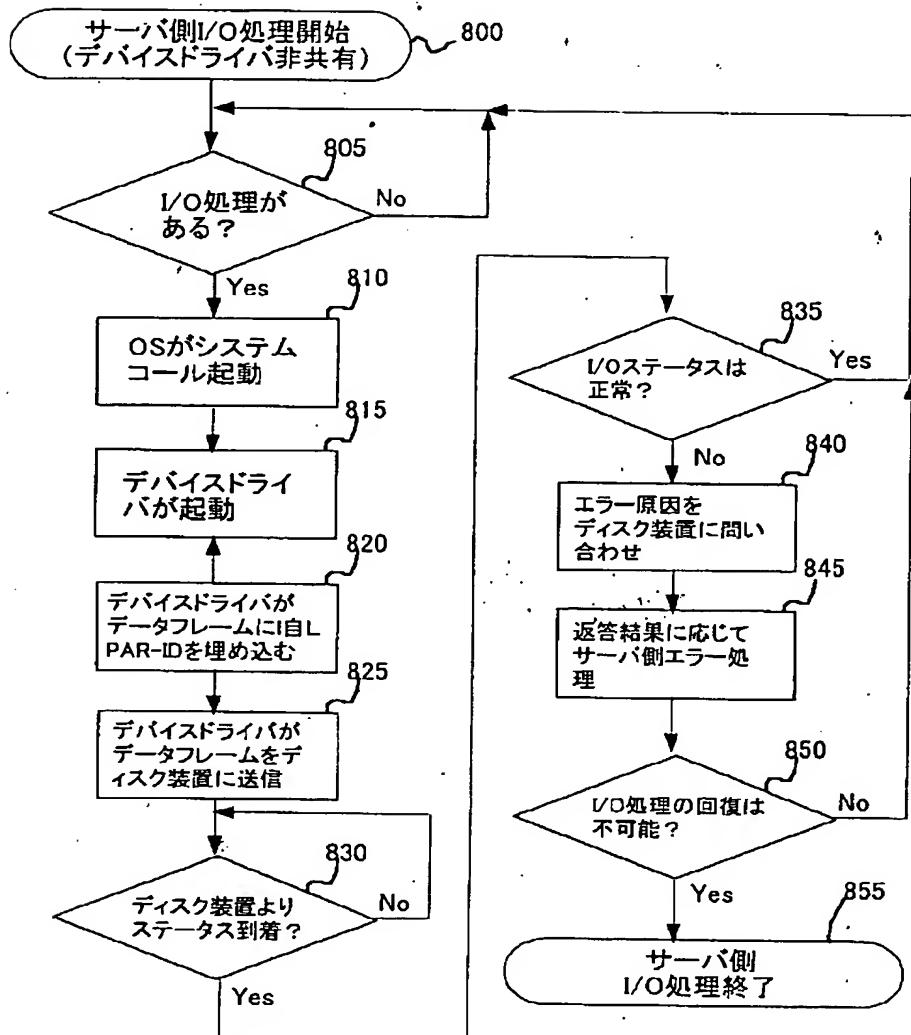


【図8】



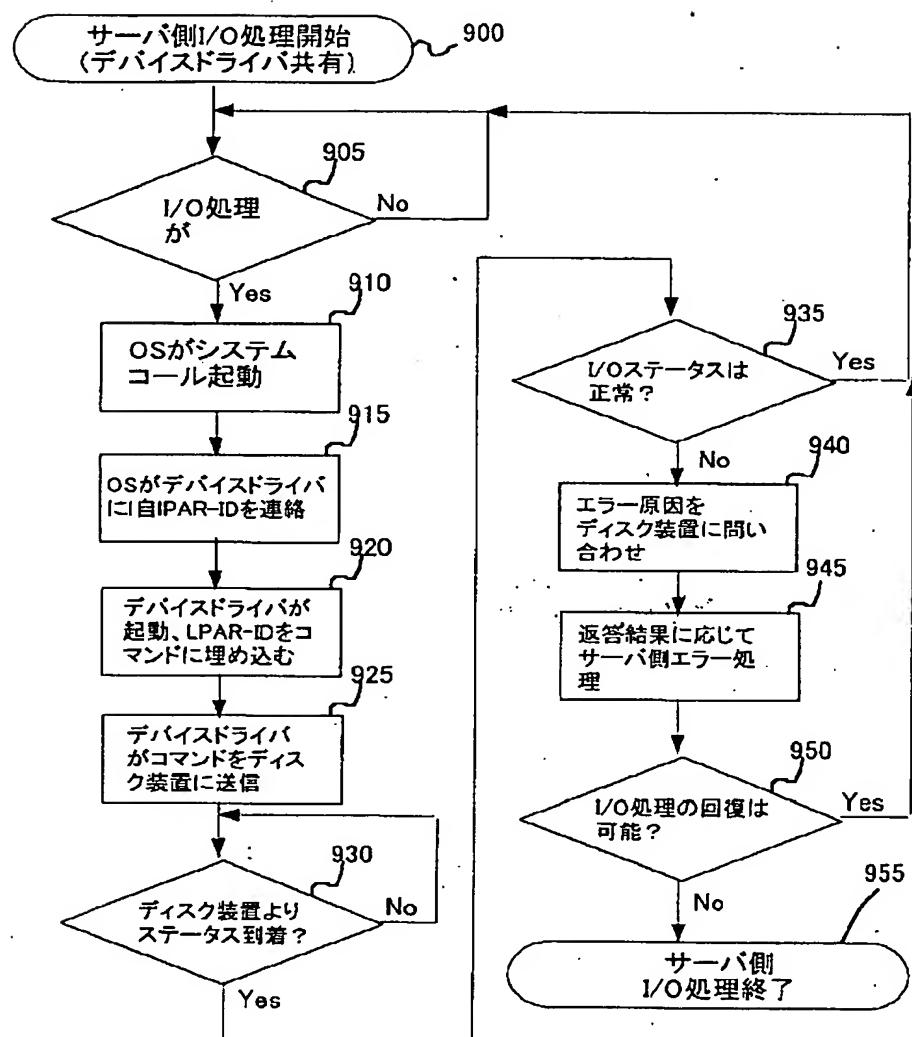
【図9】

図9



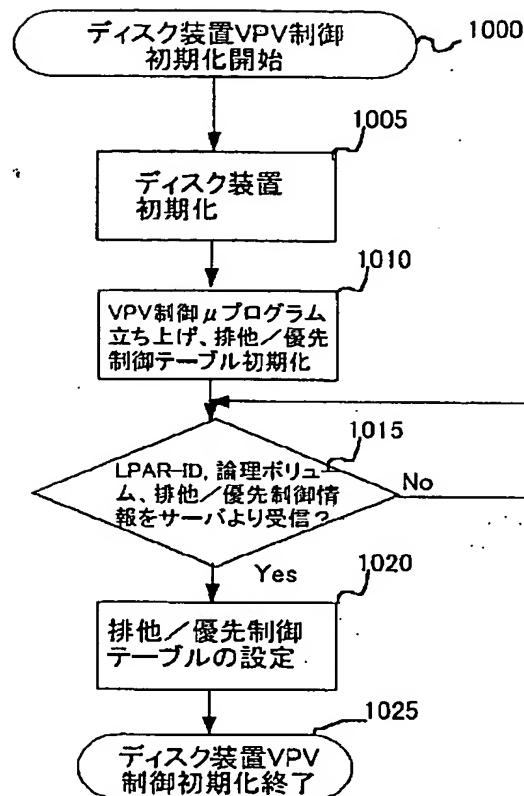
【図10】

図10



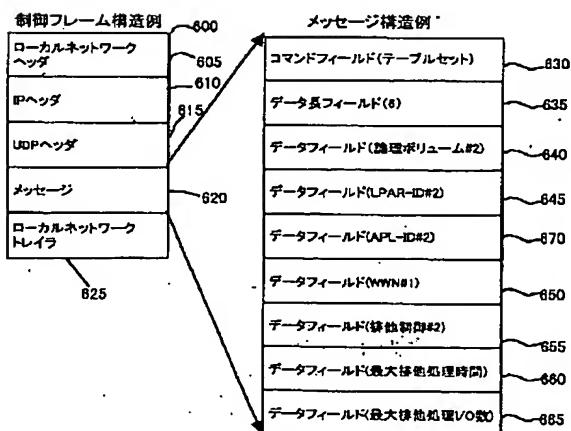
【図11】

図11



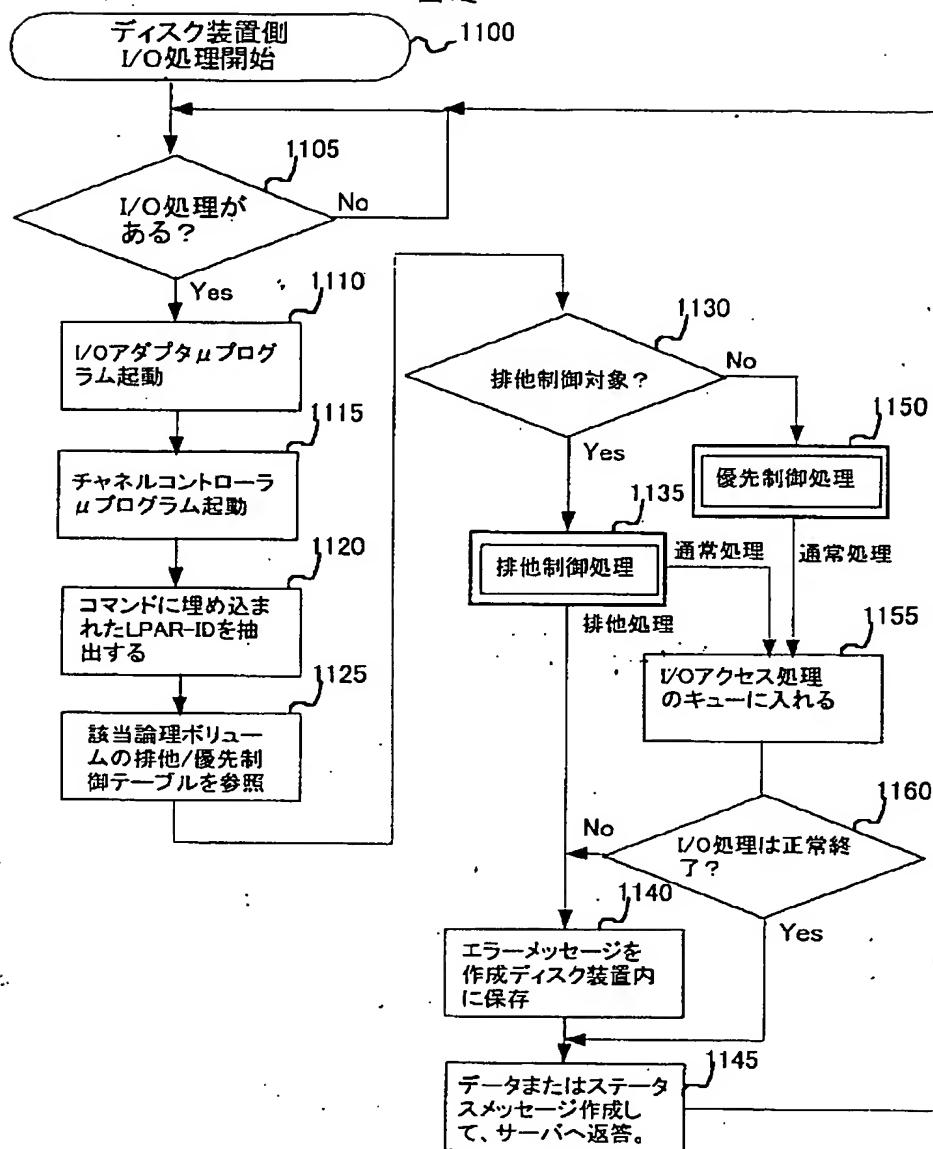
【図20】

図20



【図12】

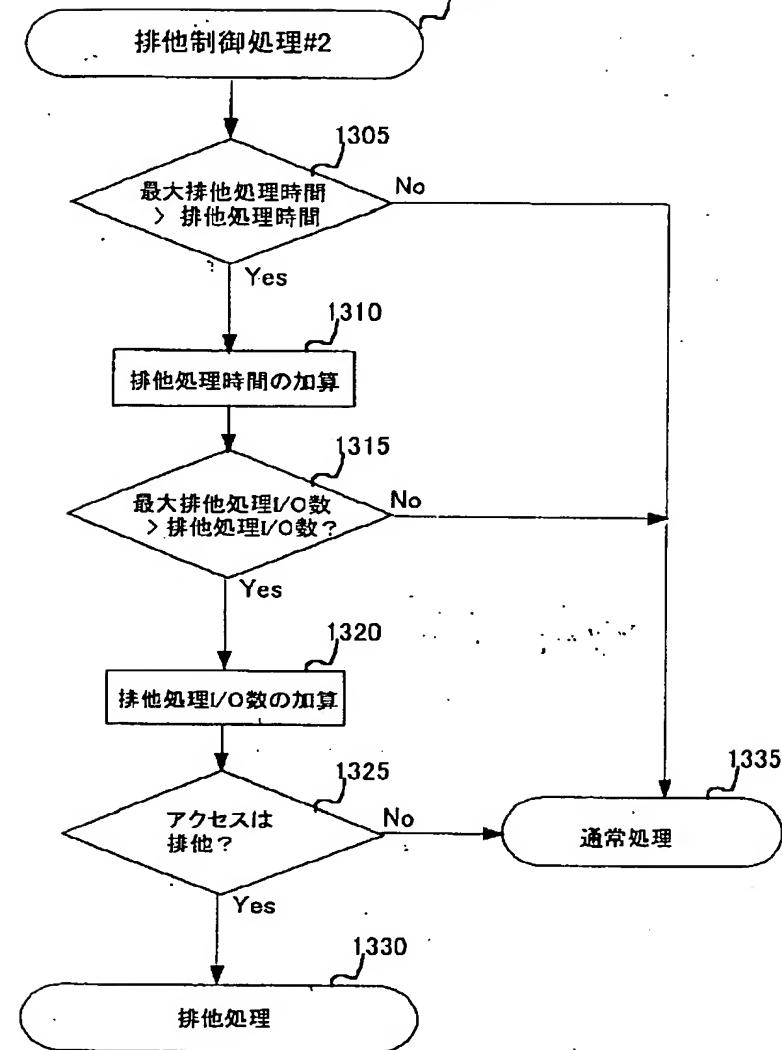
図12



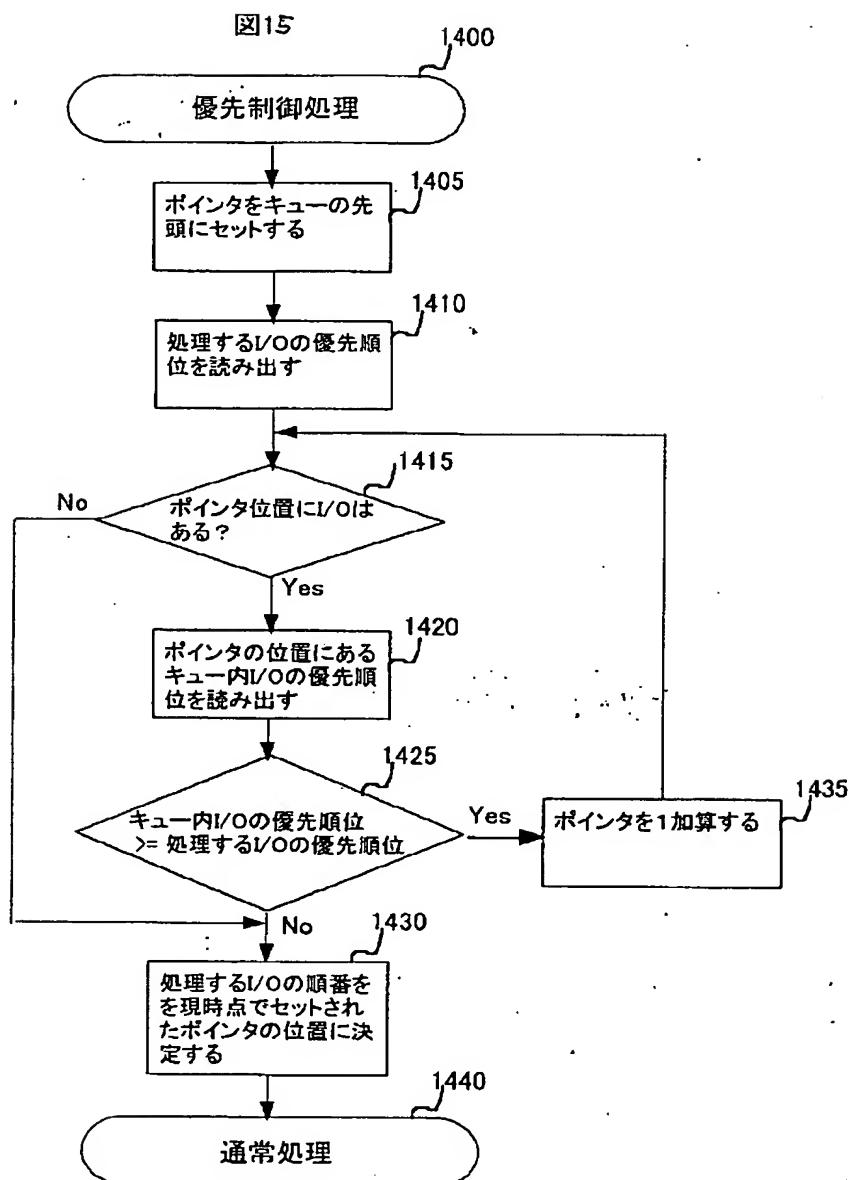
【図14】

図14-

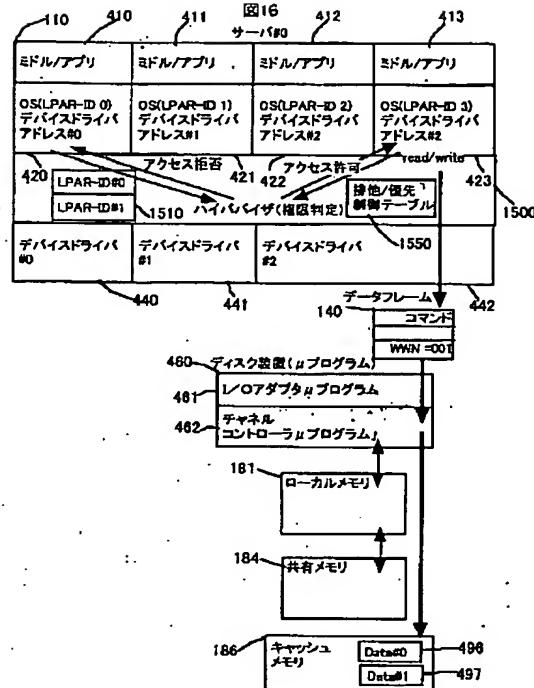
1300



【図15】

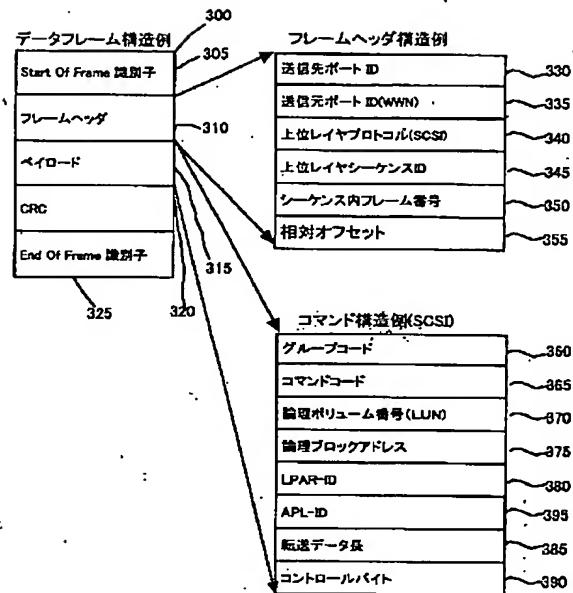


【図16】



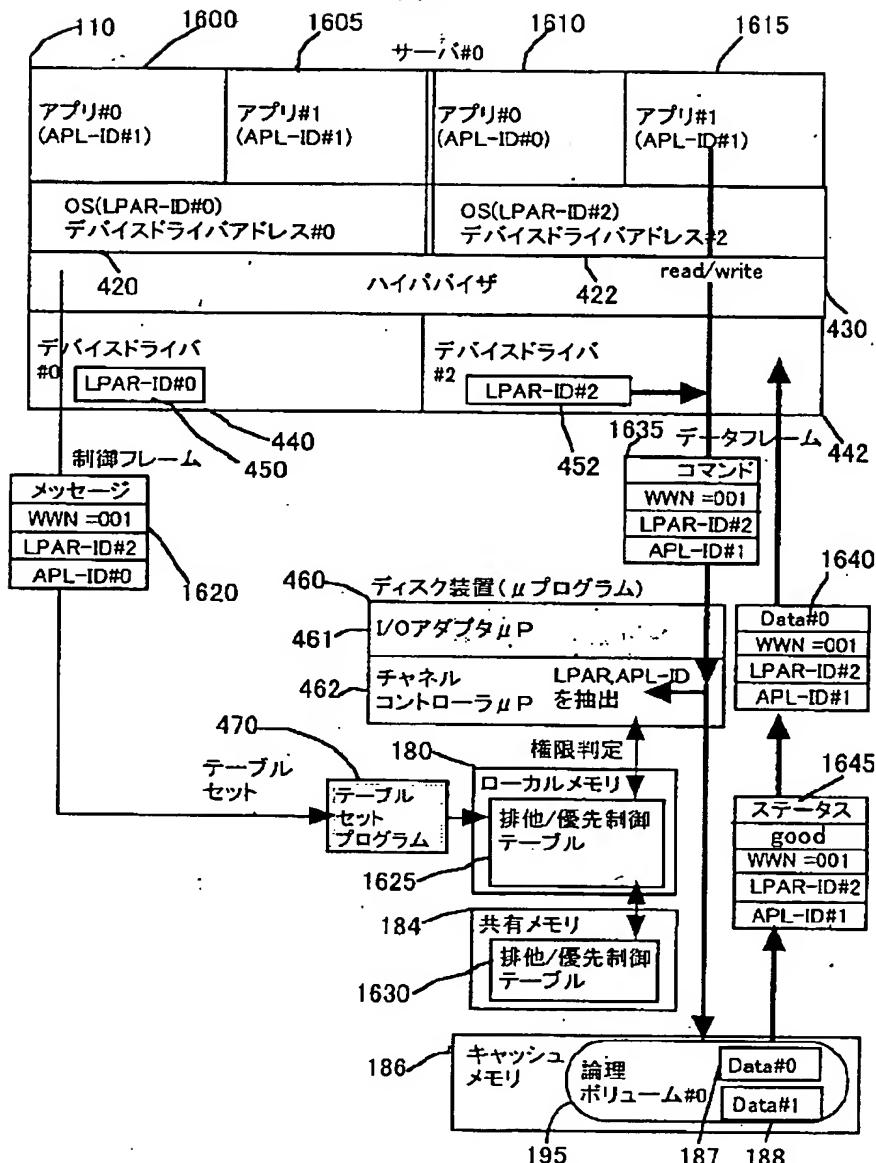
【図19】

図19



【図17】

図17



フロントページの続き

(72)発明者 小田原 宏明

東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

F ターム(参考) 5B045 EE06 EE08 EE11

5B082 EA11 JA03